

Lección 1

Necesidad de arquitectura y presentación de los microservicios



Requisitos de la empresa de cafetería



La cafetería quiere recibir informes diarios por correo electrónico sobre todos los pedidos realizados en el sitio web. Necesitan esta información para poder anticiparse a la demanda y hornear el número correcto de postres en el futuro (reduciendo el desperdicio). También quieren identificar patrones en su empresa (análisis).

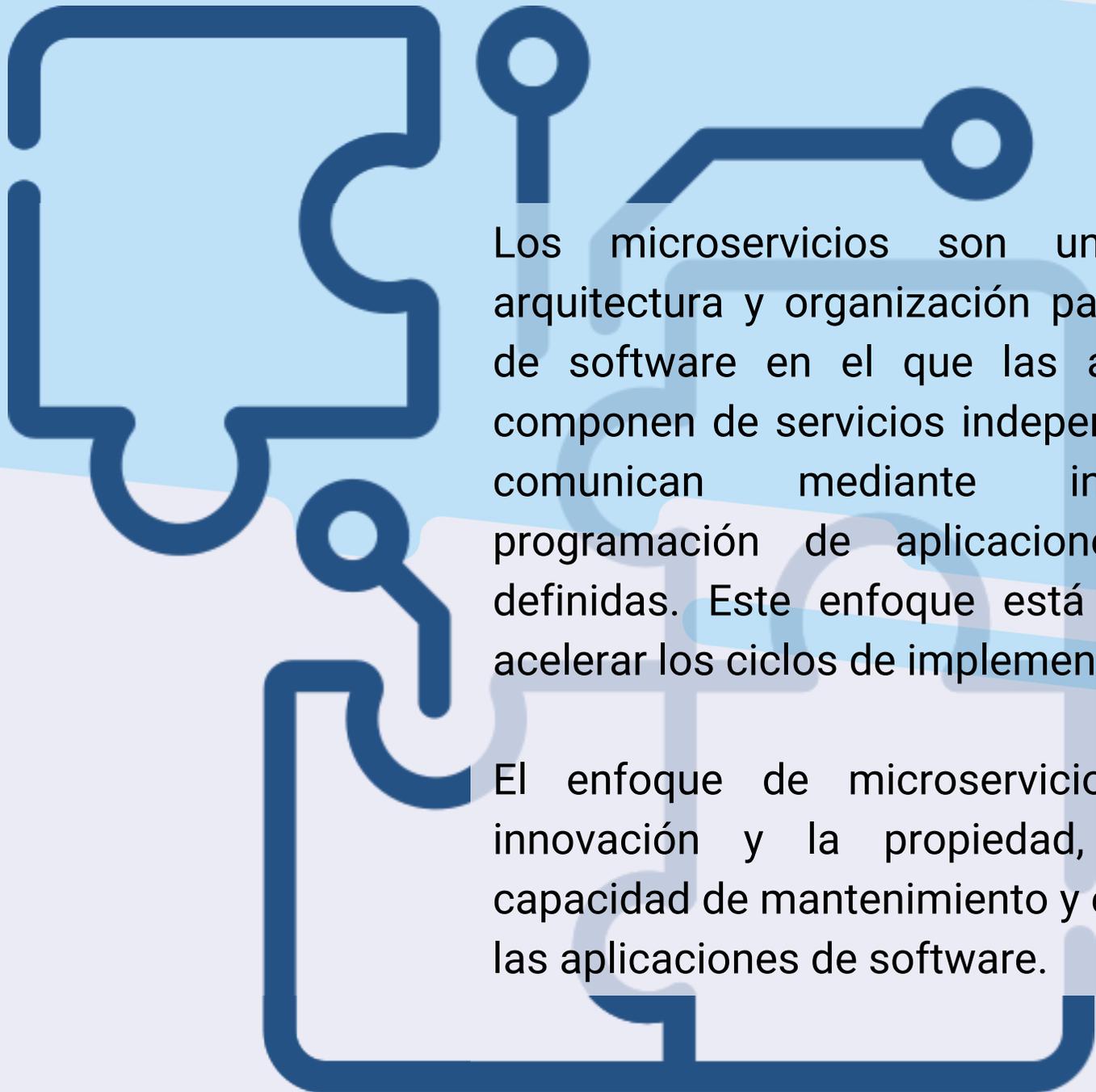
Frank y Martha quieren recibir informes diarios por correo electrónico sobre todos los pedidos realizados en el sitio web. Frank quiere anticiparse a la demanda y hornear el número correcto de postres en el futuro (reduciendo el desperdicio).

Martha quiere identificar patrones en su empresa (análisis). Actualmente, Sofía ha configurado un trabajo cron en la instancia del servidor web que envía estos mensajes de correo electrónico con el informe diario de los pedidos a Frank y Martha. Sin embargo, el trabajo cron consume muchos recursos y reduce el rendimiento del servidor web.

Olivia aconseja a Sofía y a Nikhil separar los informes de las tareas que no sean críticas para la empresa. Sofía y Nikhil quieren desacoplar más la arquitectura y trasladar el trabajo a un entorno administrado y sin servidor, que escale bien y reduzca los costos.

¿Qué son los microservicios?

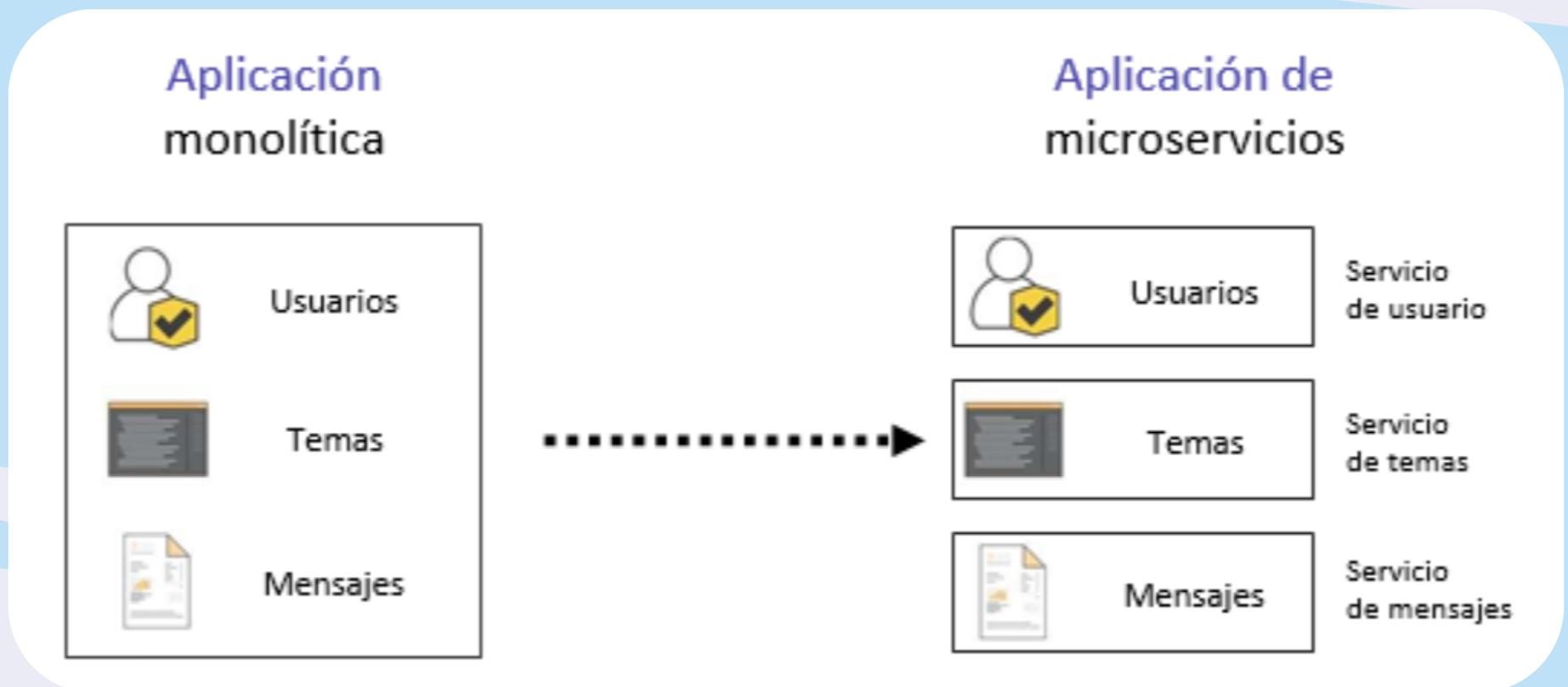
Aplicaciones compuestas por servicios independientes que se comunican a través de API bien definidas



Los microservicios son un enfoque de arquitectura y organización para el desarrollo de software en el que las aplicaciones se componen de servicios independientes que se comunican mediante interfaces de programación de aplicaciones (API) bien definidas. Este enfoque está diseñado para acelerar los ciclos de implementación.

El enfoque de microservicios fomenta la innovación y la propiedad, y mejora la capacidad de mantenimiento y escalabilidad de las aplicaciones de software.

Aplicaciones monolíticas frente a aplicaciones de microservicios



Para comprender los beneficios de los microservicios, primero consideremos una aplicación monolítica.

En el ejemplo de la izquierda, los tres procesos (usuarios, temas y mensajes) de una aplicación monolítica de foro están estrechamente acoplados. Se ejecutan como un solo servicio.

Si un proceso de una aplicación experimenta un pico de demanda, se debe escalar toda la arquitectura. Agregar o mejorar funciones se vuelve más complejo a medida que crece la base de código, lo que limita la experimentación y dificulta la implementación de nuevas ideas. La disponibilidad de las aplicaciones monolíticas también está en riesgo porque muchos procesos dependientes y con un acoplamiento ajustado aumentan el impacto de un error de un solo proceso.

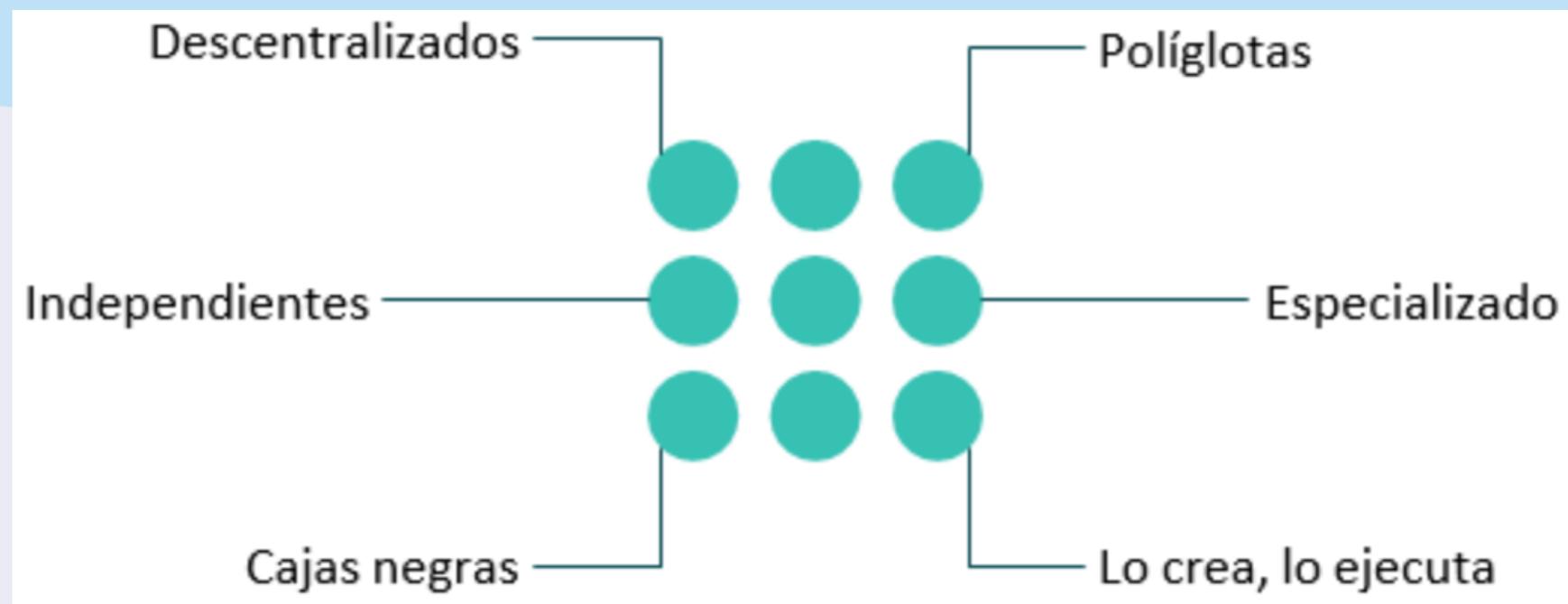
Supongamos ahora que la misma aplicación se ejecuta en una arquitectura de microservicios. Cada proceso de la aplicación se desarrolla como un componente independiente que se ejecuta como un servicio. Los servicios se comunican mediante operaciones ligeras de la API. Cada servicio realiza una única función que puede admitir múltiples aplicaciones. Debido a que los servicios se ejecutan de forma independiente, se pueden actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación.



Una arquitectura de microservicios proporciona una iteración, automatización y agilidad general mucho más rápidas. Iniciar rápido, encontrar errores rápido y recuperarse rápido.

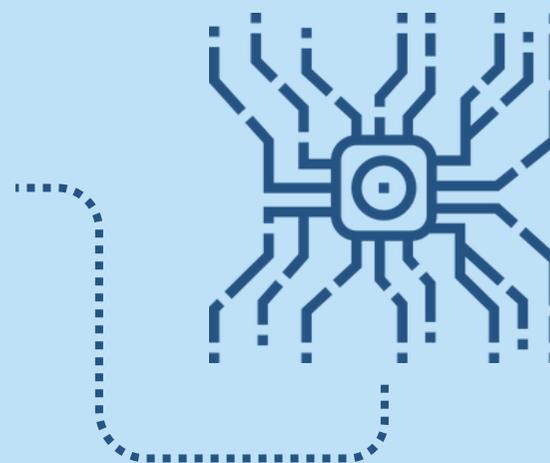
Para ver la información general sobre los microservicios en AWS, consulte [¿Qué son los microservicios?](#)

Características de los microservicios



Los microservicios comparten algunas características comunes:

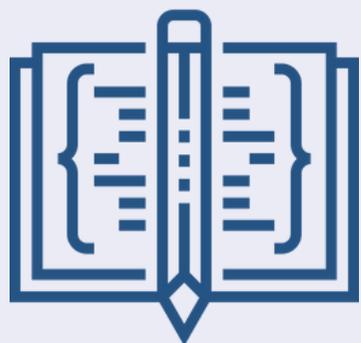
- **Descentralizados:** las arquitecturas de microservicios son sistemas distribuidos con administración de datos descentralizada. No dependen de un esquema unificado en una base de datos central. Cada microservicio tiene su propia visión sobre los modelos de datos. Los microservicios también se desarrollan, implementan, administran y operan de forma descentralizada.



- **Independientes:** cada servicio de componente de una arquitectura de microservicios puede modificarse, actualizarse o reemplazarse de forma independiente sin que afecte la función de los demás servicios. Los servicios no necesitan compartir ninguno de sus códigos o implementaciones con otros servicios. Del mismo modo, los equipos responsables de los distintos microservicios pueden funcionar de forma independiente entre ellos.



- **Especializados:** cada servicio de componente está diseñado para un conjunto de funciones y se enfoca en resolver un dominio específico. Si el código de un servicio de componente determinado alcanza un cierto nivel de complejidad, el servicio puede dividirse en dos o más servicios.



- **Políglotas:** los microservicios no siguen un solo enfoque. Los equipos tienen la libertad de elegir la mejor herramienta para sus problemas específicos. Por lo tanto, las arquitecturas de microservicios adoptan un enfoque heterogéneo en los sistemas operativos, lenguajes de programación, almacenes de datos y herramientas. Este enfoque se denomina persistencia y programación políglota.

- **Cajas negras:** los servicios de componentes individuales se diseñan como cajas negras, lo que significa que los detalles de su complejidad están ocultos para otros componentes. Cualquier comunicación entre servicios ocurre a través de las API bien definidas para evitar dependencias implícitas y ocultas.



- **Lo crea, lo ejecuta:** DevOps es un principio organizativo clave para los microservicios, donde el equipo responsable de crear un servicio, también es responsable de operarlo y mantenerlo en producción.

Entre los puntos clave de esta Lección, se incluyen los siguientes:

- Las aplicaciones de microservicios se componen de servicios independientes que se comunican a través de API bien definidas
- Los microservicios comparten las siguientes características:
 - Descentralizados: los microservicios se desarrollan, implementan, administran y operan de forma descentralizada
 - Independientes: cada servicio de componente en una arquitectura de microservicios se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios
 - Especializados: cada servicio está diseñado para un conjunto de funciones y se enfoca en resolver un problema específico
 - Políglotas: las arquitecturas de microservicios adoptan un enfoque heterogéneo en los sistemas operativos, lenguajes de programación, almacenes de datos y herramientas
 - Cajas negras: los detalles de la complejidad de los microservicios están ocultos para otros componentes
 - Lo crea, lo ejecuta: DevOps es un principio organizativo clave para los microservicios